WHAT IS CLAIMED IS:

1. A method for identifying data state anomalies capable of causing a catastrophic failure in a continuously operating software system, said method comprising the steps of:

injecting a first data state anomaly into the software system;

running the software system after the data state anomaly has been injected;

checking for an unacceptable output from the software system; and

logging the unacceptable output if an unacceptable output is observed.

2. The method of claim 1, wherein the continuously operating software system comprises a safety-critical system and the unacceptable output comprises a hazardous output.

3. The method of claim 1, wherein the continuously operating software system comprises a web site system.

4. The method of claim 1, wherein the unacceptable output comprises an undesired output.

5. The method of claim 1, wherein the unacceptable output comprises an unacceptable performance property.

6. The method of claim 1, wherein the unacceptable output comprises an unsafe shutdown of the software system.

7. The method of claim 1, further comprises the step of stopping the software system if an unacceptable output is observed.

8. The method of claim 1, further comprising repeating each of the steps using a second data state anomaly, said second data state anomaly different than the first data state anomaly.

9. The method of claim 1, further comprising the step of stopping the software system if a pre-determined period has elapsed without an unacceptable behavior being observed.

10. The method of claim 9, wherein the determined period comprises a time period.

11. The method of claim 9, wherein the determined period comprises a predetermined number of iterations of the software system.

12. The method of claim 1, further comprising the step of inserting an assertion into the software system.

13. The method of claim 12, further comprising the step of inserting a corrective action into the software system, said corrective action comprising a response to the assertion.

14. A method for estimating a safe operating period for a continuously running software system, said method comprising the steps of:

initializing the software system;

running the software system for a first pre-determined period;

pausing the software system;

injecting a first data state anomaly into the software system;

running the software system after the data state anomaly has been injected;

checking for an unacceptable output from the software system;

stopping the software system and logging the unacceptable output if an unacceptable output is observed; and

stopping the software system if a pre-determined period has elapsed without an unacceptable behavior being observed.

15. The method of claim 14, further comprising repeating each of the steps using a second data state anomaly, said second data state anomaly different than the first data state anomaly.

16. The method of claim 15, wherein the first pre-determined period is changed prior to repeating the steps.

17. The method of claim 14, wherein the first and second pre-determined periods comprise time periods.

18. The method of claim 14, wherein the first and second pre-determined periods comprise iterations of the software system.

19. The method of claim 14, further comprising the step of inserting an assertion into the software system.

20. The method of claim 14, wherein the step of logging the hazardous output comprises writing a plurality of information to a log file.

21. The method of claim 20, wherein the plurality of information comprises a time stamp.

22. The method of claim 20, wherein the plurality of information comprises an iteration count.

23. The method of claim 20, wherein the plurality of information comprises the first data state anomaly.

24. The method of claim 20, wherein the plurality of information comprises a time stamp and the first data state anomaly.

25. The method of claim 20, wherein the plurality of information comprises an iteration count and the first data state anomaly.

26. The method of claim 20, further comprising the step of analyzing the plurality of information in the log file to determine a safe operating period for the continuously operating software system.